



遷移手冊

金蝶Apusic分布式缓存 V2.0.5

版權聲明

本文件所涉及的軟件著作權、版權等知識產權已依法進行了註冊，由金蝶天燕雲計算股份有限公司合法擁有。受《中華人民共和國著作權法》《計算機軟件保護條例》《知識產權保護條例》和相關國際版權條約、法律、法規以及其它知識產權法律和條約的保護。未經授權許可，不得非法使用。

免責聲明

本文件包含的版權資訊由金蝶天燕雲計算股份有限公司合法擁有，受法律的保護，金蝶天燕雲計算股份有限公司對本文件可能涉及到的非金蝶天燕雲計算股份有限公司的資訊不承擔任何責任。在法律允許的範圍內，您可以查閱並僅能夠在《中華人民共和國著作權法》規定的合法範圍內複製和打印本文件。任何單位和個人未經金蝶天燕雲計算股份有限公司書面授權許可，不得使用、修改、再發佈本文件的任何部分和內容，否則將被視為侵權，金蝶天燕雲計算股份有限公司有依法追究其責任的權利。

本文件如有更新，不另行通知。對本文件中的問題您可向金蝶天燕雲計算股份有限公司告知或查詢。未經本公司明確授予的任何權利均予保留。

商標聲明

 是深圳市金蝶天燕雲計算股份有限公司向中華人民共和國國家商標局申請註冊的註冊商標，註冊商標專用權由金蝶天燕合法擁有，受法律保護。未經金蝶天燕的書面許可，任何單位及個人不得以任何方式或理由對該商標的任何部分進行使用、複製、修改、傳播、抄錄或與其它產品捆綁使用銷售。凡侵犯金蝶天燕商標權的，金蝶天燕將依法追究其法律責任。本文件提及的其他所有商標或註冊商標，由各自的所有人擁有。

目錄

- 1 前言
 - 1.1 適用對象
 - 1.2 相關文檔
 - 1.3 技術支持
- 2 概述
- 3 AMDC V2.0.2及以下版本遷移至V2.0.5操作說明
 - 3.1 準備工作
 - 3.1.1 環境檢查
 - 3.1.2 工具與文件準備
 - 3.1.2.1 必備工具
 - 3.1.2.2 待遷移文件收集
 - 3.2 單機模式遷移
 - 3.2.1 遷移前提
 - 3.2.2 準備工作
 - 3.2.2.1 工具與配置前置
 - 3.2.2.2 備份數據並停止AMDC V2.0.2及以下版本單節點（全量停機）
 - 3.2.3 遷移步驟
 - 3.2.3.1 配置文件遷移
 - 3.2.3.2 緩存數據遷移
 - 3.2.3.3 啟動AMDC V2.0.5服務
 - 3.2.3.4 驗證啟動
 - 3.2.4 注意事項
 - 3.3 主從模式遷移
 - 3.3.1 遷移前提
 - 3.3.2 準備工作
 - 3.3.2.1 工具與配置前置
 - 3.3.2.2 備份數據並停止AMDC V2.0.2及以下的本主從集群（全量停機）
 - 3.3.3 遷移步驟（按「主節點->所有從節點」順序執行）
 - 3.3.3.1 主節點遷移
 - 3.3.3.2 從節點遷移（所有從節點同步執行）
 - 3.3.4 注意事項
 - 3.4 哨兵模式遷移

- 3.4.1 遷移前提
- 3.4.2 準備工作
 - 3.4.2.1 工具與配置前置
 - 3.4.2.2 備份數據並全量停止AMDC V2.0.2及以下版本的哨兵 + 主從集群
- 3.4.3 遷移步驟（按「AMDC V2.0.5主從集群->AMDC V2.0.5哨兵集群」順序執行）
 - 3.4.3.1 重建AMDC V2.0.5主從集群（核心，同上一章完整步驟）
 - 3.4.3.2 哨兵節點遷移（所有哨兵節點同步執行）
- 3.4.4 注意事項
- 3.5 集群模式遷移
 - 3.5.1 遷移前提
 - 3.5.2 準備工作
 - 3.5.2.1 工具與配置前置
 - 3.5.2.2 統一環境與記錄源拓撲
 - 3.5.2.3 備份數據並全量停止AMDC V2.0.2及以下版本的集群（禁止任何寫入）
 - 3.5.2.4 數據文件預處理（所有源節點執行）
 - 3.5.3 遷移步驟（按「配置轉換->AMDC V2.0.5集群初始化->分片數據導入->集群狀態驗證」執行）
 - 3.5.3.1 初始化AMDC V2.0.5空集群
 - 3.5.3.1.1 驗證所有節點啟動成功（日誌顯示Ready to accept connections）
 - 3.5.3.2 分片數據導入
 - 3.5.3.3 集群狀態修復與數據校驗
 - 3.5.4 注意事項
- 4 Redis 8.6及以下版本遷移至AMDC V2.0.5操作說明
 - 4.1 準備工作
 - 4.1.1 環境檢查
 - 4.1.2 工具與文件準備
 - 4.2 單機模式遷移
 - 4.2.1 遷移前提
 - 4.2.2 準備工作
 - 4.2.2.1 數據與配置備份
 - 4.2.2.2 數據備份並停止Redis單節點
 - 4.2.3 遷移步驟
 - 4.2.3.1 緩存數據遷移
 - 4.2.3.2 修改redis.conf配置

- 4.2.3.3 啟動AMDC服務
- 4.2.3.4 驗證啟動
- 4.2.4 注意事項
- 4.3 主從模式遷移
 - 4.3.1 遷移前提
 - 4.3.2 準備工作
 - 4.3.2.1 工具與文件準備
 - 4.3.2.2 數據備份並停止Redis主從集群（全量停機）
 - 4.3.3 遷移步驟（按「主節點->所有從節點」順序執行）
 - 4.3.3.1 緩存數據遷移
 - 4.3.3.2 配置文件遷移（redis.conf）
 - 4.3.3.3 啟動AMDC主從節點服務
 - 4.3.3.4 主從關係驗證
 - 4.3.4 注意事項
- 4.4 哨兵模式遷移
 - 4.4.1 遷移前提
 - 4.4.2 準備工作
 - 4.4.2.1 工具與文件準備
 - 4.4.2.2 數據備份並全量停止Redis哨兵 + 主從集群
 - 4.4.3 遷移步驟
 - 4.4.3.1 緩存數據遷移
 - 4.4.3.2 配置文件遷移（redis.conf）
 - 4.4.3.3 啟動AMDC主從節點服務
 - 4.4.3.4 啟動AMDC哨兵節點服務
 - 4.4.3.5 哨兵功能驗證
 - 4.4.4 注意事項
- 4.5 集群模式遷移
 - 4.5.1 遷移前提
 - 4.5.2 準備工作
 - 4.5.2.1 工具與文件準備
 - 4.5.2.2 數據備份並全量停止Redis集群（禁止任何寫入）
 - 4.5.3 遷移步驟
 - 4.5.3.1 數據文件預處理（所有源節點執行）
 - 4.5.3.2 配置文件遷移（redis.conf, node.conf）

- 4.5.3.3 啟動AMDC集群節點
- 4.5.3.4 集群狀態驗證
- 4.5.4 注意事項
- 5 常見問題排查
 - 5.1 工具無法運行
 - 5.2 校驗或修復文件失敗
 - 5.3 amdc-cli 無法連接AMDC服務
- 6 總結
 - 6.1 AMDC V2.0.2及以下版本向AMDC V2.0.5遷移流程
 - 6.2 Redis 8.6及以下版本向AMDC V2.0.5遷移流程

1 前言

本文檔為金蝶Apusic分佈式緩存（AMDC）V2.0.5的遷移手冊，詳細介紹了如何將Redis遷移到AMDC，以及AMDC老版本遷移到V2.0.5版本。

1.1 適用對象

本文檔適用於AMDC產品運維工程師、IT系統運維工程師、開發工程師、軟件架構師及研發經理等人員。

1.2 相關文檔

了解更多AMDC V2.0.5產品相關的信息，請參閱以下AMDC V2.0.5產品手冊文檔集：

序號	手冊文檔	說明
1	金蝶Apusic分佈式緩存 V2.0.5 快速使用手冊	簡單介紹了如何快速上手使用AMDC。
2	金蝶Apusic分佈式緩存 V2.0.5 安裝手冊	詳細介紹如何在各操作系統上安裝AMDC，以及AMDC服務啟停操作，產品的註冊過程。
3	金蝶Apusic分佈式緩存 V2.0.5 緩存核心用戶手冊	詳細介紹 AMDC 相關功能的使用、配置、管理及配套工具的使用方法。
4	金蝶Apusic分佈式緩存 V2.0.5 管控台用戶手冊	詳細介紹AMDC管控台相關功能的使用和操作說明。
5	金蝶Apusic分佈式緩存 V2.0.5 開發手冊	詳細介紹基於各開發語言進行AMDC客戶端應用開發的說明。
6	金蝶Apusic分佈式緩存 V2.0.5 遷移手冊	詳細介紹AMDC歷史版本遷移升級到V2.0.5版本的說明，以及Redis遷移到AMDC的說明。
7	金蝶Apusic分佈式緩存 V2.0.5 運維手冊	詳細介紹AMDC的監控、運維、安全加固等運維說明。
8	金蝶Apusic分佈式緩存 V2.0.5 性能優化手冊	詳細介紹AMDC性能調優的說明。

1.3 技術支持

AMDC產品提供全面的技術支持服務，您可以通過以下方式獲得技術支持：

- 網址: www.apusic.com
- 電話: 400-855-5800
- 郵箱: support@apusic.com
- 金蝶雲社區: <https://vip.kingdee.com/?productId=73&productLineId=14&lang=zh-CN>

您在取得技術支持時，請提供如下信息：

1. 您的姓名
2. 公司信息與聯繫方式
3. 操作系統及其版本
4. 產品版本號
5. 出現異常及錯誤的日誌、截圖等詳細信息

2 概述

本手冊介紹兩種遷移至AMDC V2.0.5的操作說明：

1. **AMDC V2.0.2及以下版本遷移至AMDC V2.0.5**：使用amdc-conf-conv轉換AMDC V2.0.2及以下版本的配置文件為AMDC V2.0.5兼容格式，復用原有配置及數據文件，啟動後驗證功能即可；
2. **Redis 8.6及以下版本遷移至AMDC V2.0.5**：核心依託AMDC可直接加載Redis配置（redis.conf等）及數據文件（RDB/AOF等）的特性。

3 AMDC V2.0.2及以下版本遷移至V2.0.5操作說明

3.1 準備工作

3.1.1 環境檢查

- 已安裝AMDC V2.0.5（需與工具版本匹配），且安裝目錄權限正常（可讀可執行）；
- 待遷移的AMDC V2.0.2及以下版本的服務可正常訪問，或已獲取其RDB緩存文件、配置文件（YAML格式）；
- 遷移目標服務器（部署的AMDC V2.0.5服務器）具備足夠的磁盤空間，確保能存儲遷移後的緩存數據及配置文件；

3.1.2 工具與文件準備

3.1.2.1 必備工具

- [amdc-check-rdb](#)：用於校驗和修復AMDC V2.0.2及以下版本的RDB緩存文件（隨AMDC V2.0.5安裝包部署，位於/amdc/目錄下）；
- [amdc-check-aof](#)：用於校驗和修復AMDC V2.0.2及以下版本的AOF文件（同目錄部署）；
- [amdc-conf-conv](#)：用於轉換AMDC V2.0.2及以下版本的配置文件為AMDC V2.0.5兼容格式（同目錄部署）；

3.1.2.2 待遷移文件收集

- AMDC V2.0.2及以下版本的配置文件：服務端配置（如conf.yaml）或哨兵配置（如sentinel.go.yaml）；
- AMDC V2.0.2及以下版本的RDB快照文件：默認名為dump.rdb，路徑通常在安裝目錄下（需確認實際路徑）；
- 若AMDC V2.0.2及以下版本啟用了AOF持久化，則優先使用AOF文件作為遷移數據源，RDB文件作為備份遷移數據源；

3.2 單機模式遷移

單機模式遷移適用於僅部署單台AMDC服務的場景，核心流程為「文件準備 -> 配置轉換 -> 數據校驗修復 -> 部署啟動 -> 驗證」，操作簡單且無依賴關係，具體步驟如下：

3.2.1 遷移前提

- 目標服務器已安裝AMDC V2.0.5，安裝目錄權限正常（可讀可執行）；

- 已收集AMDC V2.0.2及以下版本的配置文件（如conf.yaml）、RDB快照文件(dump.rdb)，若啟用AOF持久化則優先獲取AOF文件；
- 目標服務器磁盤空間充足，至少為源數據文件大小的2倍（預留備份與運行空間）。

3.2.2 準備工作

3.2.2.1 工具與配置前置

- [amdc-check-rdb](#)：用於校驗和修復AMDC V2.0.2及以下版本的RDB緩存文件（隨AMDC V2.0.5安裝包部署，位於/amdc/目錄下）；
- [amdc-check-aof](#)：用於校驗和修復AMDC V2.0.2及以下版本的AOF文件（同目錄部署）；
- [amdc-conf-conv](#)：用於轉換AMDC V2.0.2及以下版本的配置文件為AMDC V2.0.5兼容格式（同目錄部署）；

3.2.2.2 備份數據並停止AMDC V2.0.2及以下版本單節點（全量停機）

- 停止AMDC V2.0.2及以下版本單節點的寫入操作後，再次執行bgsave命令，生成最新RDB文件；備份源端AMDC配置；
- 必須先停止AMDC V2.0.2及以下版本的單節點，禁止任何數據寫入，確保源數據文件最終一致性。

#節點執行：停止AMDC V2.0.2及以下版本的AMDC進程

```
ps -ef | grep amdc-server | grep -v grep | awk '{print $2}' | xargs kill -9
```

#驗證進程已停止（無輸出則表示成功）

```
ps -ef | grep amdc-server | grep -v grep
```

3.2.3 遷移步驟

3.2.3.1 配置文件遷移

使用 [amdc-conf-conv](#) 工具將AMDC V2.0.2及以下版本的服務端配置轉換為AMDC V2.0.5兼容格式，操作如下：

1. 備份目標服務器的AMDC V2.0.5默認配置文件（避免覆蓋丟失）：

```
cp /amdc/amdc.yaml /amdc/amdc.yaml.bak
```

2. 執行配置轉換（指定--server參數，適配服務端配置）：

```
amdc-conf-conv --server --go-yaml /path/to/conf.yaml --c-yaml
/amdc/amdc.yaml
```

3. 驗證轉換結果：打開/amdc/amdc.yaml，確認核心配置項（如端口、密碼、數據目錄、內存限制等）已正確繼承配置，無語法錯誤；

3.2.3.2 緩存數據遷移

1. 校驗並修復源RDB或AOF文件（優先處理AOF文件，數據更完整）：

#若使用AOF文件

```
amdc-check-aof /path/to/appendonlydir/appendonly.aof.manifest
```

#若使用RDB文件

```
amdc-check-rdb /path/to/dump.rdb
```

#若文件損壞，執行修復

```
amdc-check-aof --fix
```

```
/path/to/appendonlydir/appendonly.aof.1.base.rdb /
```

```
amdc-check-aof --fix
```

```
/path/to/appendonlydir/appendonly.aof.1.base.aof /
```

```
amdc-check-aof --fix
```

```
/path/to/appendonlydir/appendonly.aof.1.incr.aof
```

```
amdc-check-rdb --fix /path/to/dump.rdb
```

2. 複製修復後的文件到AMDC V2.0.5的安裝目錄：

#查看AMDC V2.0.5數據目錄（在amdc.yaml中通過dir配置項確認，默認路徑如下），複製RDB文件

```
cp /path/to/dump.rdb/ amdc/
```

#若使用AOF文件，需先啟用AOF配置（在amdc.yaml中設置appendonly yes），再複製

```
cp /path/to/appendonlydir /amdc/
```

3.2.3.3 啟動AMDC V2.0.5服務

1. 啟動AMDC V2.0.5服務，加載轉換後的配置文件：

```
amdc-server /amdc/amdc.yaml
```

2. 驗證服務啟動狀態：

```
#查看服務進程
ps -ef | grep amdc-server

#查看日誌（默認日誌路徑 / amdc/server.log）
tail -100f /amdc/server.log
```

3.2.3.4 驗證啟動

1. 連接AMDC V2.0.5服務，驗證數據完整性：

```
#使用amdc-cli連接（默認端口6359，若配置密碼需加-a選項）
amdc-cli -p 6359 -a test123

#查看鍵空間統計信息
info keyspace
```

2. 驗證功能可用性：執行寫入、修改、刪除等操作，確認服務正常響應，無配置或數據相關報錯：

3.2.4 注意事項

- 遷移過程中需停止AMDC V2.0.2及以下版本的服務，避免數據寫入導致源文件損壞；
- 遷移後觀察服務運行 10-20 分鐘，確認無日誌報錯等異常。

3.3 主從模式遷移

主從模式遷移適配下，AMDC V2.0.2及以下版本與AMDC V2.0.5無法跨版本建立主從的場景，核心原則為全量停機 -> 先遷主節點 -> 後遷從節點 -> 數據一致驗證，通過RDB或AOF全量數據導入完成遷移，全程需短暫停機（無跨版本同步環節），具體步驟如下：

3.3.1 遷移前提

- 主、從目標服務器均已安裝AMDC V2.0.5，安裝目錄權限正常（可讀、可寫、可執行），且主從節點網絡互通（無防火牆攔截6359端口）；
- 已收集AMDC V2.0.2及以下版本主、從節點的配置文件（conf.yaml）、AOF文件（優先，數據更完整）或RDB快照文件，並做好全量備份；
- 提前告知業務方停機窗口，建議選擇業務低峰期操作，停機時長約為「數據文件拷貝 + 服務啟動」耗時（根據數據量調整，一般5-30分鐘）；
- 目標服務器磁盤空間充足，至少為源數據文件大小的2倍（預留備份與運行空間）。

3.3.2 準備工作

3.3.2.1 工具與配置前置

- [amdc-check-rdb](#)：用於校驗和修復AMDC V2.0.2及以下版本的RDB緩存文件（隨AMDC V2.0.5安裝包部署，位於/amdc/目錄下）；
- [amdc-check-aof](#)：用於校驗和修復AMDC V2.0.2及以下版本的AOF文件（同目錄部署）；
- [amdc-conf-conv](#)：用於轉換AMDC V2.0.2及以下版本的配置文件為AMDC V2.0.5兼容格式（同目錄部署）；

3.3.2.2 備份數據並停止AMDC V2.0.2及以下的本主從集群（全量停機）

- 停止AMDC V2.0.2及以下版本單節點的寫入操作後，再次執行bgsave命令，生成最新RDB文件；備份源端AMDC配置；
- 必須先停止所有AMDC V2.0.2及以下版本的主、從節點，禁止任何數據寫入，確保源數據文件最終一致性。

#主節點 + 所有從節點統一執行：停止AMDC V2.0.2及以下版本的AMDC進程

```
ps -ef | grep amdc-server | grep -v grep | awk '{print $2}' | xargs kill -9
```

#驗證進程已停止（無輸出則表示成功）

```
ps -ef | grep amdc-server | grep -v grep
```

3.3.3 遷移步驟（按「主節點->所有從節點」順序執行）

3.3.3.1 主節點遷移

主節點為數據核心，需優先完成配置轉換、數據校驗導入和服務啟動，作為後續從節點的同步源。

步驟1：配置文件轉換與優化

1. 備份AMDC V2.0.5主節點默認配置（避免覆蓋）：

```
cp /amdc/amdc.yaml /amdc/amdc-master.yaml.bak
```

2. 將AMDC V2.0.2及以下版本的主節點配置轉換為AMDC V2.0.5主節點專用配置（命名為amdc-master.yaml，便於區分）：

```
amdc-conf-conv --server --go-yaml /path/to/conf.yaml --c-yaml  
/amdc/amdc-master.yaml
```

3. 檢查AMDC V2.0.5主節點配置，啟用主節點身份（無replicaof配置），確認核心配置項：

```
vi /amdc/amdc-master.yaml
```

關鍵配置確認 / 修改：

- port:6359（端口統一，若需修改則所有節點同步）；
- requirepass: 你的密碼（若源主節點有密碼，需保留，後續從節點需配置masterauth）；
- dir:/amdc/（數據目錄，與實際規劃一致）；
- appendonly:yes（建議啟用AOF，與源端保持一致，若源端僅用RDB則設為no）；
- logfile:amdc-master.log（主節點獨立日誌，便於排查）。

步驟2：數據文件校驗、修復與導入

1. 校驗並修復源AMDC V2.0.2及以下版本的主節點數據文件（主節點執行，若文件無損壞則跳過修復）：

#若使用AOF文件（推薦）

```
amdc-check-aof /path/to/appendonlydir/appendonly.aof.manifest
```

#若使用RDB文件

```
amdc-check-rdb /path/to/dump.rdb
```

#若文件損壞，執行修復

```
amdc-check-aof --fix
```

```
/path/to/appendonlydir/appendonly.aof.1.base.rdb /
```

```
amdc-check-aof --fix
/path/to/appendonlydir/appendonly.aof.1.base.aof /
amdc-check-aof --fix
/path/to/appendonlydir/appendonly.aof.1.incr.aof

amdc-check-rdb --fix /path/to/dump.rdb
```

2. 將修復後的乾淨數據文件拷貝到AMDC V2.0.5主節點數據目錄，並修改權限：

```
#拷貝AOF文件（推薦，若用RDB則執行下一行）
cp /path/to/appendonlydir/ /amdc/

#拷貝RDB文件
cp /path/to/dump.rdb /amdc/

#賦權（避免服務無權限讀取）
chmod 644 /amdc/{appendonlydir/, dump.rdb}
```

步驟3：啟動AMDC V2.0.5主節點服務

```
#啟動主節點，指定專用配置文件
amdc-server /amdc/amdc-master.yaml

#驗證主節點啟動狀態
ps -ef | grep amdc-server | grep master # 查看進程
tail -100f /amdc/amdc-master.log # 查看日誌
```

日誌顯示 Ready to accept connections 表示主節點啟動成功；

連接主節點驗證數據完整性（核心，確保數據導入成功）：

```
amdc-cli -p 6359 -a 密碼 # 連接主節點

info keyspace # 查看鍵空間統計（確認數據量）
```

3.3.3.2 從節點遷移（所有從節點同步執行）

從節點無需單獨導入數據，啟動後通過AMDC V2.0.5內部主從同步，從已啟動的AMDC V2.0.5主節點拉取全量數據，確保與主節點數據完全一致。

步驟1：配置文件轉換與主從關聯

1. 備份AMDC V2.0.5從節點默認配置：

```
cp /amdc/amdc.yaml /amdc/amdc-slave.yaml.bak
```

2. 將AMDC V2.0.2以下版本從節點配置轉換為AMDC V2.0.5從節點專用配置（命名為 amdc-slave.yaml）：

```
amdc-conf-conv --server --go-yaml /path/to/conf.yaml --c-yaml  
/amdc/amdc-slave.yaml
```

3. 檢查AMDC V2.0.5從節點配置，指定主節點信息，啟用從節點身份：

```
vi /amdc/amdc-slave.yaml
```

關鍵配置確認 / 修改：

- port:6360（端口統一，若需修改則所有節點同步）；
- requirepass: 你的密碼（若源主節點有密碼，需保留，後續從節點需配置 masterauth）；
- dir:/amdc/（數據目錄，與實際規劃一致）；
- appendonly:yes（建議啟用AOF，與源端保持一致，若源端僅用RDB則設為no）；
- replicaof: 主節點IP主節點Port # 指向AMDC V2.0.5主節點的IP和端口；
- logfile:amdc-slave.log（主節點獨立日誌，便於排查）。

步驟2：啟動AMDC V2.0.5從節點服務

無需向從節點數據目錄拷貝任何源數據文件，AMDC V2.0.5從節點會自動從主節點同步：

```
#啟動從節點，指定專用配置文件  
amdc-server /amdc/amdc-slave.yaml  
  
#驗證從節點啟動狀態
```

```
ps -ef | grep amdc-server    # 查看進程
tail -100f /amdc/amdc-slave.log  # 查看日誌
```

日誌顯示 MASTER <-> REPLICA sync: Finished with success 表示從節點已成功連接主節點並開始同步。

步驟3：驗證從節點同步狀態

連接從節點，確認同步完成且數據與主節點一致：

```
amdc-cli -p 6360 -a 密碼 # 連接從節點

info replication # 查看同步狀態
```

關鍵狀態確認：

- role:slave：從節點身份正常；
- master_host：主節點IP、masterport:6359：主節點指向正確；
- master_link_status: up：主從連接正常。

3.3.4 注意事項

- 遷移全程必須先全量停止AMDC V2.0.2及以下版本的主從節點，禁止數據寫入，否則會導致源數據文件不一致，導入後數據丟失；
- 主節點數據導入後必須先驗證數據完整性，再啟動從節點，避免從節點同步錯誤數據；
- 從節點無需拷貝任何源數據文件，由AMDC V2.0.5內部主從同步完成，手動拷貝會導致同步衝突；
- 主、從節點的核心配置必須一致（AOF開關、端口、密碼等），否則會導致連接失敗或同步異常；
- 遷移後觀察集群運行10-20分鐘，確認主從同步無延遲、日誌無報錯、業務訪問正常。

3.4 哨兵模式遷移

AMDC V2.0.2及以下版本與AMDC V2.0.5無法跨版本主從同步、哨兵無法監控跨版本節點的場景，核心原則為全量停機->先遷主從節點（同第六章）->後遷哨兵節點->重新關聯監控，通過「AMDC V2.0.5主從集群重建 + 哨兵重新監控」完成遷移，全程需短暫業務停機，具體步驟如下：

3.4.1 遷移前提

- 所有主從節點、哨兵節點均已安裝AMDC V2.0.5，網絡互通（開放6359服務端口、26359哨兵端口）；

- 已收集AMDC V2.0.2及以下版本的主從節點配置、數據文件、哨兵配置文件(sentinel_go.yaml)，並做好全量備份；
- 提前規劃停機窗口（低峰期），停機時長為「AMDC V2.0.2及以下版本停機 + AMDC V2.0.5主從啟動 + 哨兵啟動」耗時（一般10-40分鐘）；
- 所有目標節點時間同步，磁盤空間充足（主節點數據目錄預留2倍源數據空間）。

3.4.2 準備工作

3.4.2.1 工具與配置前置

- [amdc-check-rdb](#)：用於校驗和修復AMDC V2.0.2及以下版本的RDB緩存文件（隨AMDC V2.0.5安裝包部署，位於/amdc/目錄下）；
- [amdc-check-aof](#)：用於校驗和修復AMDC V2.0.2及以下版本的AOF文件（同目錄部署）；
- [amdc-conf-conv](#)：用於轉換AMDC V2.0.2及以下版本的配置文件為AMDC V2.0.5兼容格式（同目錄部署）；

3.4.2.2 備份數據並全量停止AMDC V2.0.2及以下版本的哨兵 + 主從集群

- 停止AMDC V2.0.2及以下版本單節點的寫入操作後，再次執行bgsave命令，生成最新RDB文件；備份源端AMDC配置；
- 必須先停止所有AMDC V2.0.2及以下版本的哨兵節點和主從節點，禁止數據寫入和哨兵監控，確保源數據一致性。

#所有節點統一執行：停止AMDC V2.0.2及以下版本的AMDC主從 + 哨兵進程

```
ps -ef | grep -E 'amdc-server | amdc-sentinel' | grep -v grep | awk '{print $2}' | xargs kill -9
```

#驗證進程已停止（無輸出則成功）

```
ps -ef | grep -E 'amdc-server | amdc-sentinel' | grep -v grep
```

3.4.3 遷移步驟（按「AMDC V2.0.5主從集群->AMDC V2.0.5哨兵集群」順序執行）

3.4.3.1 重建AMDC V2.0.5主從集群（核心，同上一章完整步驟）

- 嚴格按照上一章主從模式遷移的所有步驟，完成AMDC V2.0.5主從集群的搭建；
- 遷移AMDC V2.0.5主節點：配置轉換 -> 數據校驗導入 -> 啟動 -> 驗證數據完整性；
- 遷移所有AMDC V2.0.5從節點：配置轉換（關聯主節點） -> 啟動 -> 驗證主從同步；

- 完成主從集群整體驗證：數據一致性、主從功能、日誌無異常。

此步驟為哨兵遷移的基礎，必須確保AMDC V2.0.5主從集群完全正常運行，再進行後續哨兵遷移。

3.4.3.2 哨兵節點遷移（所有哨兵節點同步執行）

哨兵節點無需導入數據，僅需將AMDC V2.0.2及以下版本的哨兵配置轉換為AMDC V2.0.5兼容格式，並修改監控目標為已搭建完成的AMDC V2.0.5主節點。

步驟1：配置文件轉換與監控目標修改

1. 備份AMDC V2.0.5哨兵默認配置：

```
cp /amdc/sentinel.yaml /amdc/sentinel-c.yaml.bak
```

2. 將AMDC V2.0.2及以下版本的哨兵配置轉換為AMDC V2.0.5哨兵專用配置（命名為sentinel-c.yaml）：

```
amdc-conf-conv --sentinel --go-yaml /path/to/sentinel_go.yaml --c-yaml /amdc/sentinel.yaml
```

3. 檢查AMDC V2.0.5哨兵配置，確認哨兵關鍵配置：

```
vi /amdc/sentinel.yaml
```

關鍵配置修改或確認（核心為monitor配置項，指向AMDC V2.0.5主節點）：

#哨兵核心監控配置（替換為AMDC V2.0.5主節點信息，其餘參數與源端保持一致）

SENTINEL：

```
sentinel monitor: mymaster 10.0.1.100 (示例IP, 請替換為實際IP) 6359 2 #
mymaster: 監控名稱（與源端一致），後接C主節點IP、端口、quorum閾值
```

```
sentinel down-after-milliseconds: mymaster 30000 # 主觀下線超時時間（與源端一致）
```

```
sentinel failover-timeout: mymaster 180000 # 故障轉移超時時間（與源端一致）
```

```
sentinel auth-pass: mymaster 密碼 # 與C主節點的requirepass一致，無密碼則省略
```

其他配置確認：

- port:26359（哨兵默認端口，與源端一致）；
- logfile:sentinel.log（哨兵獨立日誌）；
- dir:/amdc/（哨兵數據目錄，存儲監控狀態）。

```
#所有哨兵節點同步啟動，指定專用配置文件
amdc-sentinel /amdc/sentinel.yaml

#驗證哨兵啟動狀態
ps -ef | grep amdc-sentinel # 查看進程
tail -100f /amdc/sentinel.log # 查看日誌
```

步驟2：啟動AMDC V2.0.5哨兵服務

日誌顯示 Sentinel ID is xxxxxxxxxxxxxxxxxxxxxx 表示哨兵啟動成功。

關鍵驗證：哨兵已成功發現AMDC V2.0.5主從集群的所有節點

```
#連接哨兵端口（26359），查詢監控狀態

amdc-cli -p 26359

info sentinel # 查看哨兵監控信息

sentinel slaves mymaster # 查看監控的從節點列表
```

驗證結果要求：

- 從節點列表中顯示的IP和端口均為AMDC V2.0.5從節點信息，狀態為online。

3.4.4 注意事項

- 遷移前必須全量停止AMDC V2.0.2及以下版本的哨兵 + 主從節點，避免哨兵持續監控已停機節點，或數據寫入導致源文件不一致；
- 必須先完成AMDC V2.0.5主從集群的搭建和驗證，再啟動哨兵節點，否則哨兵會監控不到有效節點，導致遷移失敗；
- 哨兵配置中監控名稱（如mymaster）、quorum閾值、超時時間建議與源端保持一致，減少客戶端配置修改；
- 所有哨兵節點的監控配置必須完全一致（監控目標、密碼、閾值等），否則會導致哨兵集群分裂，故障轉移異常；
- 遷移後觀察10-20分鐘，重點監控：哨兵日誌無報錯、主從切換無異常、同步延遲在可接受範圍、業務訪問無中斷。

3.5 集群模式遷移

集群模式遷移適配下，AMDC V2.0.2及以下版本與AMDC V2.0.5無法跨版本建立集群、無法跨版本同步分片數據的場景，核心原則為全量停機->分片數據全量導出->AMDC V2.0.5集群初始化->分片數據批量導入->集群驗證，因集群為多主多從分片架構，無跨版本同步可能，需通過「全量數據導出-導入」完成整集群遷移，全程需業務停機，具體步驟如下：

3.5.1 遷移前提

- 目標集群所有節點（與源集群節點數一致：M主N從）均已安裝AMDC V2.0.5，網絡互通（開放6359服務端口、16359集群gossip通信端口）；
- 已收集AMDC V2.0.2及以下版本的集群所有節點的配置文件、AOF、RDB數據文件，並記錄源集群拓撲信息（執行cluster nodes 獲取槽位分配、主從關係）；
- 提前規劃長停機窗口（根據數據量調整，建議低峰期），停機時長為「數據導出 + 集群初始化 + 數據導入 + 驗證」耗時（數據量10G內約30-60分鐘）；
- 所有目標節點磁盤空間充足（每節點預留2倍源節點數據空間），時間同步，防火牆放行6359、16359端口。

3.5.2 準備工作

3.5.2.1 工具與配置前置

- [amdc-check-rdb](#)：用於校驗和修復AMDC V2.0.2及以下版本的RDB緩存文件（隨AMDC V2.0.5安裝包部署，位於/amdc/目錄下）；
- [amdc-check-aof](#)：用於校驗和修復AMDC V2.0.2及以下版本的AOF文件（同目錄部署）；
- [amdc-conf-conv](#)：用於轉換AMDC V2.0.2及以下版本的配置文件為AMDC V2.0.5兼容格式（同目錄部署）；

3.5.2.2 統一環境與記錄源拓撲

1. 所有目標節點創建統一目錄並賦權：
2. 觀察AMDC V2.0.2及以下版本的集群核心拓撲：

連接AMDC V2.0.2及以下版本的集群任意節點，執行以下命令，保存輸出結果（後續AMDC V2.0.5集群初始化完全復用）：

```
amdc-cli cluster nodes # 記錄：主從節點IP和端口、節點ID、槽位分配（如0-5460）
```

或者查看node.conf文件，其中記錄有集群節點IP，端口，節點ID，槽位分配等信息。

```
29755a26cc60c056b3af39eb66dc5ccbfbdd20c1 127.0.0.1:6383@16383 slave
882b57434ef8c361fcea44c327379880181af1a6 0 1769505692837 2 connected

882b57434ef8c361fcea44c327379880181af1a6 127.0.0.1:6380@16380 master
- 0 1769505692432 2 connected 5461-10922

980c1df981902c2df617ca81d4df7219b1e86a8b 127.0.0.1:6384@16384
myself,slave 2793d6ad527eee582aa30890d7c8407e9c112b4f003 connected

43d25d48fac16f9b8f6b519f88a889179b8a6e87 127.0.0.1:6382@16382 slave
ac9f0c10e742b4570f6436db8af281c1d51ea570 0 1769505692331 1 connected

2793d6ad527eee582aa30890d7c8407e9c112b4f 127.0.0.1:6381@16381 master
- 0 1769505692332 3 connected 10923-16383

ac9f0c10e742b4570f6436db8af281c1d51ea570 127.0.0.1:6379@16379 master
- 0 1769505692228 1 connected 0-5460
```

記錄每個AMDC V2.0.2及以下版本的每個主節點IP與端口號，備份對應機器上的AOF或RDB文件用於後續將數據遷移到AMDC V2.0.5集群：

3.5.2.3 備份數據並全量停止AMDC V2.0.2及以下版本的集群（禁止任何寫入）

- 停止AMDC V2.0.2及以下版本單節點的寫入操作後，再次執行bgsave命令，生成最新RDB文件；備份源端AMDC配置；

```
#所有AMDC V2.0.2及以下版本的集群節點統一執行：停止進程
ps -ef | grep amdc-server | grep -v grep | awk '{print $2}' | xargs
kill -9

#驗證進程已停止（無輸出則成功）
ps -ef | grep amdc-server | grep -v grep
```

3.5.2.4 數據文件預處理（所有源節點執行）

對每個AMDC V2.0.2及以下版本的主節點的AOF或RDB文件進行校驗 + 修復，確保為乾淨數據文件，避免導入AMDC V2.0.5後出現數據損壞：

```
#優先使用AOF文件（每個節點單獨處理，集群為分片存儲，各節點數據獨立）
amdc-check-aof /path/to/nodeX/appendonlydir/appendonly.aof.manifest

#若無AOF則使用RDB文件
amdc-check-rdb /path/to/nodeX/dump.rdb

#若文件損壞，執行修復
amdc-check-aof --fix
/path/to/appendonlydir/appendonly.aof.1.base.rdb /
amdc-check-aof --fix
/path/to/appendonlydir/appendonly.aof.1.base.aof /
amdc-check-aof --fix
/path/to/appendonlydir/appendonly.aof.1.incr.aof

amdc-check-rdb --fix /path/to/nodeX/dump.rdb
```

3.5.3 遷移步驟（按「配置轉換->AMDC V2.0.5集群初始化->分片數據導入->集群狀態驗證」執行）

3.5.3.1 初始化AMDC V2.0.5空集群

步驟1：所有AMDC V2.0.5節點配置轉換與集群啟用

1. 配置轉換：

```
amdc-conf-conv --server --go-yaml /path/to/nodeX/conf.yaml --c-yaml
```

```
/amdc/amdc.yaml
```

2. 檢查配置文件：

```
vi /amdc/amdc.yaml
```

關鍵配置：

- port:6359（集群服務端口，與源端一致）；
- logfile: amdc-cluster-XXX.log（節點獨立日誌）；
- dir:/amdc/（統一數據目錄）；
- requirepass: 密碼（與源端一致，所有節點相同）；
- cluster-enabled:yes（啟用集群模式）；
- cluster-config-file:nodes-XXX.conf（每個節點單獨的集群配置文件）；
- cluster-node-timeout:15000（集群節點超時時間，與源端一致）；
- cluster-require-full-coverage:yes（開啟全槽位覆蓋檢查）。

3. 所有節點配置完成後，啟動所有AMDC V2.0.5節點（僅啟動服務，未形成集群）：

```
amdc-server /amdc/amdc-cluster-XXX.yaml
```

3.5.3.1.1 驗證所有節點啟動成功（日誌顯示Ready to accept connections)

步驟2：執行集群初始化（槽位 + 主從分配）

選擇任意一個AMDC V2.0.5主節點，執行amdc-cli --cluster create命令，嚴格按照源拓撲指定節點IP、端口和槽位分配，示例以3主3從（源拓撲槽位 0-5460、5461-10922、10923-16383）為例：

```
#命令格式: amdc-cli --cluster create 主1 主2 主3 從1 從2 從3 --cluster-replicas 1
```

```
#--cluster-replicas 1: 表示1主1從，與源端一致
```

```
amdc-cli --cluster create 10.0.1.10 (示例IP, 請替換為實際IP) :6359
10.0.1.11 (示例IP, 請替換為實際IP) :6359 192.168.1.12:6359
192.168.1.13:6359 192.168.1.14:6359 192.168.1.15:6359 --cluster-
replicas 1
```

執行成功後，驗證集群初始化狀態：

```
amdc-cli -p 6359 -a 密碼 cluster nodes
```

要求：全部槽位均已合理分配至集群各節點，所有節點狀態為 myself，master 或 slave，無 fail 狀態。

3.5.3.2 分片數據導入

先啟動AMDC V2.0.5單機節點用於加載RDB或AOF文件，再使用AMDC V2.0.5 amdc-cli --cluster import 命令，將單機節點中的數據自動重新映射導入到集群中，命令為集群標準化導入方式，無需手動操作數據文件，避免文件權限、格式不兼容等問題。

```
#核心命令格式（一行執行，換行符\僅為排版）
```

```
amdc-cli --cluster import 集群主節點IP: 集群主節點Port \
--cluster-from 單節點IP: 單節點Port \
--cluster-copy \
--cluster-replace \
-a 集群密碼
```

參數詳解：

- --cluster-copy：必加參數，複製數據時保留源單節點的原始數據，避免源數據被刪除；
- --cluster-replace：可選參數，若集群中存在與源節點同名的鍵，直接覆蓋（建議遷移時添加，確保數據與源端一致，根據業務需求選擇）。

因集群為分片存儲，每個AMDC V2.0.2及以下版本的源分片僅存儲對應槽位的數據，需按「源分片主節點->AMDC V2.0.5對應分片主節點」的順序，逐個分片執行導入命令，全程在AMDC V2.0.5集群任意可操作節點執行即可，無需在目標主節點本地操作。

示例：以3主3從集群為例（源主1 -> C主1、源主2 -> C主2、源主3 -> C主3）

```
#導入分片1：源主1(192.168.1.1:6359) -> AMDC V2.0.5主1(10.0.1.10 (示例IP,
請替換為實際IP) :6359)
```

```
amdc-cli --cluster import 10.0.1.10 (示例IP, 請替換為實際IP) :6359 \
--cluster-from 192.168.1.1:6359 \
--cluster-copy \
--cluster-replace \
-a 集群密碼
```

```
#導入分片2：源主2(192.168.1.2:6359) -> AMDC V2.0.5主2(10.0.1.11 (示例IP,
請替換為實際IP) :6359)
```

```
amdc-cli --cluster import 10.0.1.11 (示例IP, 請替換為實際IP) :6359 \
--cluster-from 192.168.1.2:6359 \
--cluster-copy \
--cluster-replace \
-a 集群密碼
```

```
#導入分片3：源主3(192.168.1.3:6359) -> AMDC V2.0.5主
3(192.168.1.12:6359)
```

```
amdc-cli --cluster import 192.168.1.13:6359 \
```

```
--cluster-from 192.168.1.3:6359 \

--cluster-copy \

--cluster-replace \

-a 集群密碼
```

3.5.3.3 集群狀態修復與數據校驗

因手動導入數據，需執行集群狀態修復，確保槽位與數據匹配，無槽位丟失或數據錯亂：

```
#驗證槽位覆蓋（必須顯示16384個槽位全部分配）
amdc-cli -p 6359 -a 密碼 cluster info
cluster info 輸出中 cluster_slots_assigned:16384 表示槽位全部分配，無遺漏。
```

3.5.4 注意事項

- 集群遷移必須全量停機，且源集群所有節點禁止寫入，因集群為分片存儲，單個節點數據寫入會導致分片數據不一致；
- 集群 gossip通信端口 16359(6359+10000) 必須放行，否則節點無法發現集群拓撲，導致集群分裂；
- 執行amdc-cli --cluster import命令時，--cluster-copy為必加參數，省略該參數會直接刪除AMDC V2.0.2及以下版本的源節點的原始數據，造成不可逆的源數據丟失；--cluster-replace 建議遷移時添加，確保覆蓋AMDC V2.0.5集群內的同名測試鍵，與源端數據完全一致；
- 若源集群存在數據傾斜（某分片數據量過大），導入前需提前評估AMDC V2.0.5集群對應節點的磁盤、內存資源；遷移後及時對傾斜分片進行擴容，避免集群出現性能瓶頸、影響業務訪問速度；
- 遷移完成後建議觀察集群運行10-20分鐘，重點監控：各分片數據無丟失、主節點故障轉移自動觸發、跨分片操作正常執行、主從同步無持續延遲、業務訪問延遲符合預期。

4 Redis 8.6及以下版本遷移至AMDC V2.0.5操作說明

4.1 準備工作

4.1.1 環境檢查

- 目標服務器已安裝AMDC V2.0.5，安裝目錄權限正常（可讀可執行）；
- 已收集Redis的RDB快照文件(dump.rdb)，若啟用AOF持久化則優先獲取AOF文件；
- 目標服務器磁盤空間充足，至少為源數據文件大小的2倍（預留備份與運行空間）。

4.1.2 工具與文件準備

- [amdc-check-rdb](#)：用於校驗和修復AMDC V2.0.2及以下版本的RDB緩存文件（隨AMDC V2.0.5AMDC安裝包部署，位於/amdc/目錄下）；
- [amdc-check-aof](#)：用於校驗和修復AMDC V2.0.2及以下版本的AMDC的AOF文件（同目錄部署）；

4.2 單機模式遷移

單機模式遷移適用於僅部署單台AMDC服務的場景，核心流程為「文件準備 -> 數據校驗修復 -> 部署啟動 -> 驗證」，具體步驟如下：

4.2.1 遷移前提

- 目標服務器已安裝AMDC V2.0.5，安裝目錄權限正常（可讀可執行）；
- 已收集Redis的RDB快照文件(dump.rdb)，若啟用AOF持久化則優先獲取AOF文件；
- 目標服務器磁盤空間充足，至少為源數據文件大小的2倍（預留備份與運行空間）。

4.2.2 準備工作

4.2.2.1 數據與配置備份

- 檢查AMDC V2.0.5安裝目錄、數據存儲目錄權限（確保AMDC用戶可讀寫）；
- RDB文件和Redis配置文件複製至目標端AMDC指定備份目錄（如/amdc 644），確保AMDC可讀取。

4.2.2.2 數據備份並停止Redis單節點

- 停止業務對Redis的寫入操作，再次執行bgsave命令，生成最新RDB文件；備份源端Redis配置(redis.conf)；

- 必須先停止Redis單節點，禁止任何數據寫入，確保源數據文件最終一致性。

#節點執行：停止Redis進程

```
ps -ef | grep amdc-server | grep -v grep | awk '{print $2}' | xargs
kill -9
```

#驗證進程已停止（無輸出則表示成功）

```
ps -ef | grep amdc-server | grep -v grep
```

4.2.3 遷移步驟

4.2.3.1 緩存數據遷移

1. 校驗並修復源RDB或AOF文件（優先處理AOF文件，數據更完整）：

#若使用AOF文件

```
amdc-check-aof /path/to/appendonly.aof
```

#若使用RDB文件

```
amdc-check-rdb /path/to/dump.rdb
```

#若文件損壞，執行修復

```
amdc-check-aof --fix /path/to/appendonly.aof
```

```
amdc-check-rdb --fix /path/to/dump.rdb
```

2. 複製修復後的文件到AMDC V2.0.5的安裝目錄：

#查看AMDC V2.0.5數據目錄（在amdc.yaml中通過dir配置項確認，默認路徑如下），複製RDB文件

```
cp /path/to/dump.rdb/ amdc/
```

#若使用AOF文件，需先啟用AOF配置（在amdc.yaml中設置appendonly yes），再複製

```
cp /path/to/appendonly.aof /amdc/
```

4.2.3.2 修改redis.conf配置

1. 修改Redis服務配置文件(redis.conf)，追加以下配置項到配置文件最下方：

```
worker-threads 1
```

4.2.3.3 啟動AMDC服務

1. 啟動AMDC V2.0.5服務，直接加載Redis的配置文件：

```
amdc-server /amdc/redis.conf
```

2. 驗證服務啟動狀態：

```
#查看服務進程
ps -ef | grep amdc-server

#查看日誌（默認日誌路徑 / amdc/server.log）
tail -100f /amdc/server.log
```

4.2.3.4 驗證啟動

1. 連接AMDC V2.0.5服務，驗證數據完整性：

```
#使用amdc-cli連接（Redis默認端口6379，若配置密碼需加-a選項）
amdc-cli -p 6379 -a test123

#查看鍵空間統計信息
info keyspace
```

2. 驗證功能可用性：執行寫入、修改、刪除等操作，確認服務正常響應，無配置或數據相關報錯：

4.2.4 注意事項

- 遷移過程中需停止Redis服務，避免數據寫入導致源文件損壞；
- 遷移後觀察服務運行 10-20 分鐘，確認無日誌報錯等異常。

4.3 主從模式遷移

主從模式遷移適用於部署一主多從AMDC服務的場景，核心流程為「文件準備 -> 數據校驗修復 -> 部署啟動 -> 驗證」，具體步驟如下：

4.3.1 遷移前提

- 主、從目標服務器均已安裝AMDC V2.0.5，安裝目錄權限正常（可讀、可寫、可執行），且主從節點網絡互通（無防火牆攔截6359端口）；
- 已收集Redis主、從節點的配置文件（redis.conf）、AOF文件（優先，數據更完整）或RDB快照文件，並做好全量備份；
- 提前告知業務方停機窗口，建議選擇業務低峰期操作，停機時長約為「數據文件拷貝 + 服務啟動」耗時（根據數據量調整，一般5-30分鐘）；
- 目標服務器磁盤空間充足，至少為源數據文件大小的2倍（預留備份與運行空間）。

4.3.2 準備工作

4.3.2.1 工具與文件準備

- [amdc-check-rdb](#)：用於校驗和修復AMDC V2.0.2及以下版本的RDB緩存文件（隨AMDC V2.0.5AMDC安裝包部署，位於/amdc/目錄下）；
- [amdc-check-aof](#)：用於校驗和修復AMDC V2.0.2及以下版本的AMDC的AOF文件（同目錄部署）；

4.3.2.2 數據備份並停止Redis主從集群（全量停機）

- 停止業務對Redis的寫入操作，再次執行bgsave命令，生成最新RDB文件；備份源端Redis配置（redis.conf）；
- 必須先停止所有Redis主、從節點，禁止任何數據寫入，確保源數據文件最終一致性。

#主節點 + 所有從節點統一執行：停止Redis進程

```
ps -ef | grep amdc-server | grep -v grep | awk '{print $2}' | xargs
kill -9
```

#驗證進程已停止（無輸出則表示成功）

```
ps -ef | grep amdc-server | grep -v grep
```

4.3.3 遷移步驟（按「主節點->所有從節點」順序執行）

4.3.3.1 緩存數據遷移

1. 校驗並修復源Redis主節點數據文件（主節點執行，若文件無損壞則跳過修復）：

#若使用AOF文件（推薦）

```
amdc-check-aof /path/to/appendonly.aof
```

#若使用RDB文件

```
amdc-check-rdb /path/to/dump.rdb
```

#若文件損壞，執行修復

```
amdc-check-aof --fix /path/to/appendonly.aof
```

```
amdc-check-rdb --fix /path/to/dump.rdb
```

2. 將修復後的乾淨數據文件拷貝到AMDC V2.0.5主節點數據目錄，並修改權限：

#拷貝AOF文件（推薦，若用RDB則執行下一行）

```
cp /path/to/appendonly.aof /amdc/
```

#拷貝RDB文件

```
cp /path/to/dump.rdb /amdc/
```

#賦權（避免服務無權限讀取）

```
chmod 644 /amdc/{appendonly.aof, dump.rdb}
```

4.3.3.2 配置文件遷移 (redis.conf)

1. 將Redis主/從節點的redis.conf拷貝至AMDC主/從節點的根目錄。
2. 修改Redis服務配置文件(redis.conf)，追加以下配置項到配置文件最下方：

```
worker-threads 1
```

4.3.3.3 啟動AMDC主從節點服務

1. 先啟動AMDC主節點：切換至主節點AMDC安裝目錄，執行啟動命令，確認主節點服務啟動成功；

```
#啟動節點，指定專用配置文件
amdc-server /amdc/redis.conf

#驗證從節點啟動狀態
ps -ef | grep amdc-server    # 查看進程
tail -100f /amdc/server.log  # 查看日誌
```

2. 再啟動所有AMDC從節點：逐個啟動從節點AMDC服務，查看每個從節點的啟動日誌，確認無報錯。

```
#啟動從節點，指定專用配置文件
amdc-server /amdc/redis.conf

#驗證從節點啟動狀態
ps -ef | grep amdc-server    # 查看進程
tail -100f /amdc/server.log  # 查看日誌
```

4.3.3.4 主從關係驗證

- 在AMDC主節點執行命令：info replication命令，查看從節點列表，確認所有從節點已正常連接主節點（master_link_status為up）；
- 在每個AMDC從節點執行命令：info replication命令，確認與主節點同步正常（offset一致）；
- 執行數據讀寫測試，主節點set key，從節點get key，確認數據同步正常。

4.3.4 注意事項

- 遷移全程必須先全量停止Redis主從節點，禁止數據寫入，否則會導致源數據文件不一致，導入後數據丟失；
- 主節點數據導入後必須先驗證數據完整性，再啟動從節點，避免從節點同步錯誤數據；
- 從節點無需拷貝任何源數據文件，由AMDC V2.0.5內部主從同步完成，手動拷貝會導致同步衝突；
- 主、從節點的核心配置必須一致（AOF開關、端口、密碼等），否則會導致連接失敗或同步異常；
- 遷移後觀察集群運行10-20分鐘，確認主從同步無延遲、日誌無報錯、業務訪問正常。

4.4 哨兵模式遷移

AMDC V2.0.5支持加載Redis配置redis.conf和sentinel.conf配置文件，核心流程為全量停機->先遷主從節點->後遷哨兵節點->重新關聯監控，通過「主從集群重建 + 哨兵重新監控」完成遷移，全程需短暫業務停機，具體步驟如下：

4.4.1 遷移前提

- 所有主從節點、哨兵節點均已安裝AMDC V2.0.5，網絡互通（開放6379服務端口、26379哨兵端口）；
- 已收集Redis主從節點配置(redis.conf)、數據文件、哨兵配置文件(sentinel.conf)，並做好全量備份；
- 提前規劃停機窗口（低峰期），停機時長為「Redis停機 + AMDC V2.0.5主從啟動 + 哨兵啟動」耗時（一般10-40分鐘）；
- 所有目標節點時間同步，磁盤空間充足（主節點數據目錄預留2倍源數據空間）。

4.4.2 準備工作

4.4.2.1 工具與文件準備

- [amdc-check-rdb](#)：用於校驗和修復AMDC V2.0.2及以下版本的RDB緩存文件（隨AMDC V2.0.5AMDC安裝包部署，位於/amdc/目錄下）；
- [amdc-check-aof](#)：用於校驗和修復AMDC V2.0.2及以下版本的AMDC的AOF文件（同目錄部署）；

4.4.2.2 數據備份並全量停止Redis哨兵 + 主從集群

- 停止業務對Redis的寫入操作，再次執行bgsave命令，生成最新RDB文件；備份源端Redis配置(redis.conf)；
- 必須先停止所有Redis哨兵節點和主從節點，禁止數據寫入和哨兵監控，確保源數據一致性。

#所有節點統一執行：停止Redis主從 + 哨兵進程

```
ps -ef | grep -E 'redis-server | redis-sentinel' | grep -v grep |
awk '{print $2}' | xargs kill -9
```

#驗證進程已停止（無輸出則成功）

```
ps -ef | grep -E 'redis-server | redis-sentinel' | grep -v grep
```

4.4.3 遷移步驟

4.4.3.1 緩存數據遷移

1. 校驗並修復源Redis主節點數據文件（主節點執行，若文件無損壞則跳過修復）：

#若使用AOF文件 (推薦)

```
amdc-check-aof /path/to/appendonly.aof
```

#若使用RDB文件

```
amdc-check-rdb /path/to/dump.rdb
```

#若文件損壞，執行修復

```
amdc-check-aof --fix /path/to/appendonly.aof
```

```
amdc-check-rdb --fix /path/to/dump.rdb
```

2. 將修復後的乾淨數據文件拷貝到AMDC V2.0.5主節點數據目錄，並修改權限：

#拷貝AOF文件 (推薦，若用RDB則執行下一行)

```
cp /path/to/appendonly.aof /amdc/
```

#拷貝RDB文件

```
cp /path/to/dump.rdb /amdc/
```

#賦權 (避免服務無權限讀取)

```
chmod 644 /amdc/{appendonly.aof, dump.rdb}
```

4.4.3.2 配置文件遷移 (redis.conf)

1. 將Redis主/從節點的redis.conf拷貝至AMDC主/從節點的根目錄。
2. 修改Redis服務配置文件(redis.conf)，追加以下配置項到配置文件最下方：

```
worker-threads 1
```

3. 哨兵配置文件(sentinel.conf)無需修改，可拷貝至AMDC哨兵節點的根目錄。

4.4.3.3 啟動AMDC主從節點服務

1. 先啟動AMDC主節點：切換至主節點AMDC安裝目錄，執行啟動命令，確認主節點服務啟動成功；

```
#啟動節點，指定專用配置文件
amdc-server /amdc/redis.conf

#驗證從節點啟動狀態
ps -ef | grep amdc-server    # 查看進程
tail -100f /amdc/server.log  # 查看日誌
```

2. 再啟動所有AMDC從節點：逐個啟動從節點AMDC服務，查看每個從節點的啟動日誌，確認無報錯。

```
#啟動從節點，指定專用配置文件
amdc-server /amdc/redis.conf

#驗證從節點啟動狀態
ps -ef | grep amdc-server    # 查看進程
tail -100f /amdc/server.log  # 查看日誌
```

4.4.3.4 啟動AMDC哨兵節點服務

1. 再啟動所有AMDC哨兵節點，查看哨兵服務進程與日誌狀態，確保啟動成功。

```
#啟動節點，指定專用配置文件
amdc-sentinel /amdc/sentinel.conf

#驗證從節點啟動狀態
ps -ef | grep amdc-sentinel  # 查看進程
tail -100f /amdc/sentinel.log # 查看日誌
```

4.4.3.5 哨兵功能驗證

- 在任意AMDC哨兵節點執行命令：info sentinel命令，查看哨兵監控的主從節點數量、狀態，確認哨兵能正常識別AMDC主從節點；
- 測試業務是否能通過哨兵自動獲取新主節點地址，讀寫業務正常；測試完成後，恢復原AMDC主節點服務

4.4.4 注意事項

- 遷移前必須全量停止Redis哨兵 + 主從節點，避免哨兵持續監控已停機節點，或數據寫入導致源文件不一致；

- 拷貝sentinel.conf後，若源端與目標端AMDC主節點IP不同，必須修改sentinel monitor參數中的主節點IP，否則哨兵無法監控到目標主節點；
- 必須先啟動主從節點，再啟動哨兵節點，確保哨兵啟動時能正常發現主從節點，避免監控失敗。
- 遷移後觀察10-20分鐘，重點監控：哨兵日誌無報錯、主從切換無異常、同步延遲在可接受範圍、業務訪問無中斷。

4.5 集群模式遷移

AMDC V2.0.5支持加載Redis配置redis.conf和node.conf配置文件，核心流程為「文件準備 -> 數據校驗修復 -> 部署啟動 -> 驗證」，具體步驟如下：

4.5.1 遷移前提

- 目標集群所有節點（與源集群節點數一致：M主N從）均已安裝AMDC V2.0.5，網絡互通；
- 已收集Redis集群所有節點的配置文件、AOF、RDB數據文件，並記錄源集群拓撲信息（執行cluster nodes 獲取槽位分配、主從關係）；
- 提前規劃長停機窗口（根據數據量調整，建議低峰期），停機時長為「數據導出 + 集群初始化 + 數據導入 + 驗證」耗時（數據量10G內約30-60分鐘）；
- 所有目標節點磁盤空間充足（每節點預留2倍源節點數據空間）。

4.5.2 準備工作

4.5.2.1 工具與文件準備

- [amdc-check-rdb](#)：用於校驗和修復AMDC V2.0.2及以下版本的RDB緩存文件（隨AMDC V2.0.5AMDC安裝包部署，位於/amdc/目錄下）；
- [amdc-check-aof](#)：用於校驗和修復AMDC V2.0.2及以下版本的AMDC的AOF文件（同目錄部署）；

4.5.2.2 數據備份並全量停止Redis集群（禁止任何寫入）

- 停止業務對Redis的寫入操作，再次執行bgsave命令，生成最新RDB文件；備份源端Redis配置（redis.conf）；

#所有Redis集群節點統一執行：停止進程

```
ps -ef | grep amdc-server | grep -v grep | awk '{print $2}' | xargs kill -9
```

```
#驗證進程已停止（無輸出則成功）
```

```
ps -ef | grep amdc-server | grep -v grep
```

4.5.3 遷移步驟

4.5.3.1 數據文件預處理（所有源節點執行）

對每個Redis主節點的AOF或RDB文件進行校驗 + 修復，確保為乾淨數據文件，避免導入AMDC後出現數據損壞：

```
#優先使用AOF文件（每個節點單獨處理，集群為分片存儲，各節點數據獨立）
```

```
amdc-check-aof /path/to/nodeX/appendonly.aof
```

```
#若無AOF則使用RDB文件
```

```
amdc-check-rdb /path/to/nodeX/dump.rdb
```

```
#若文件損壞，執行修復
```

```
amdc-check-aof --fix /path/to/appendonly.aof
```

```
amdc-check-rdb --fix /path/to/nodeX/dump.rdb
```

4.5.3.2 配置文件遷移（redis.conf, node.conf）

1. 將Redis每個集群節點的redis.conf拷貝至AMDC對應集群節點的根目錄。
2. 修改Redis服務配置文件(redis.conf)，追加以下配置項到配置文件最下方：

```
worker-threads 1
```

3. 將Redis每個集群節點node.conf拷貝至AMDC對應集群節點的根目錄。

4.5.3.3 啟動AMDC集群節點

- 逐個啟動所有AMDC集群節點，執行啟動命令，查看每個節點的啟動日誌，確認無報錯（重點關注集群連接及槽位加載情況）。

```
#啟動節點，指定專用配置文件
```

```
amdc-server /amdc/redis.conf
```

```
#驗證從節點啟動狀態
```

```
ps -ef | grep amdc-server    # 查看進程  
tail -100f /amdc/server.log  # 查看日誌
```

4.5.3.4 集群狀態驗證

- 在任意AMDC節點執行命令：cluster info，確認集群狀態為ok，所有槽位分配完整、無未分配槽位；
- 執行命令：cluster slots，查看槽位分配與源端一致，每個節點負責的槽位正確；
- 執行數據讀寫測試，跨分片查詢測試，確認數據完整、業務正常；

4.5.4 注意事項

- 集群遷移必須全量停機，且源集群所有節點禁止寫入，因集群為分片存儲，單個節點數據寫入會導致分片數據不一致；
- 若源集群存在數據傾斜（某分片數據量過大），導入前需提前評估AMDC V2.0.5集群對應節點的磁盤、內存資源；遷移後及時對傾斜分片進行擴容，避免集群出現性能瓶頸、影響業務訪問速度；
- 遷移完成後建議觀察集群運行10-20分鐘，重點監控：各分片數據無丟失、主節點故障轉移自動觸發、跨分片操作正常執行、主從同步無持續延遲、業務訪問延遲符合預期。

5 常見問題排查

5.1 工具無法運行

問題1：終端輸入amdc-check-rdb提示"command not found" (Linux)；

- 原因：工具路徑未添加到系統環境變量；
- 解決：輸入完整路徑運行（如 /usr/local/amdc/amdc-check-rdb），或添加環境變量（echo "export PATH=/usr/local/amdc/:\$PATH" >> /etc/profile, source /etc/profile）。

5.2 校驗或修復文件失敗

問題1:amdc-check-rdb或amdc-check-aof 提示「Permission denied」；

- 原因：無文件讀取或寫入權限；
- 解決：使用chmod修改文件權限 (chmod 644 dump.rdb)，或使用sudo運行工具 (sudo amdc-check-rdb --fix dump.rdb)。

問題2：修復後AMDC仍無法啟動；

- 原因：文件損壞過於嚴重，或工具版本與AMDC版本不匹配；
- 解決：使用備份文件恢復，升級或降級工具版本，與AMDC服務版本一致。

5.3 amdc-cli 無法連接AMDC服務

問題1：連接提示 "Could not connect to AMDC at 127.0.0.1:6359: Connection refused"；

- 原因：AMDC服務未啟動；
- 解決：啟動AMDC服務（amdc-server /path/to/amdc.yaml, Linux）。

問題2：遠程連接提示 "Could not connect to AMDC at 10.0.1.100（示例IP，請替換為實際IP）:6359: Connection timed out"；

- 原因：防火牆未開放6359端口，或AMDC配置不允許遠程連接；
- 解決：開放防火牆端口 (firewall-cmd --add-port=6359/tcp --permanent, Linux)，修改AMDC配置 (bind 0.0.0.0, protected-mode no)，重啟AMDC服務。

問題3：連接後執行命令提示 "NOAUTH Authentication required";

- 原因：AMDC服務設置了密碼，連接時未輸入；
- 解決：連接時用 -a 選項輸入密碼，或交互模式輸入 AUTH 密碼。

6 總結

本文檔詳細介紹了AMDC V2.0.2及以下版本與Redis 8.6及以下版本向AMDC V2.0.5的適配遷移流程，核心總結如下：

6.1 AMDC V2.0.2及以下版本向AMDC V2.0.5遷移流程

- 遷移前需完成環境檢查、工具與文件準備，確保權限充足、文件完整；
- 配置文件遷移通過 `amdc-conf-conv` 工具實現，需指定配置類型、源或目標文件路徑，轉換前務必備份目標配置文件；
- 緩存數據遷移基於RDB文件，需先通過 `amdc-check-rdb` 校驗文件完整性，修復損壞文件後部署到AMDC V2.0.5安裝目錄；
- 遷移後需驗證服務啟動狀態、數據完整性和功能可用性，確保遷移成功；
- 遷移過程中若遇到工具運行、文件處理、服務啟動等問題，可參考常見問題排查章節快速解決。

6.2 Redis 8.6及以下版本向AMDC V2.0.5遷移流程

- 安裝AMDC，解壓安裝包至目標目錄，完成基礎安裝；
- 停止源端Redis服務，收集對應模式核心文件（配置文件+數據文件）並備份；
- 使用Redis的配置文件按對應模式順序啟動AMDC節點，確保無報錯；
- 驗證數據及功能正常，切換業務流量，保留源端用於回滾。

遵循本文檔操作流程，可實現版本AMDC V2.0.2及以下版本與Redis 8.6及以下版本向AMDC V2.0.5的平滑遷移，保障緩存服務的連續性和數據安全性。

全國統一服務熱線
4008-555-800

金蝶天燕雲計算股份有限公司（簡稱“金蝶天燕”）成立於2000年，前身為“金蝶中間件公司”，是金蝶集團旗下基礎軟件平台提供商，雲計算國家標準制定企業，國家信創產業核心軟件企業。金蝶天燕是國家863重點研發計劃與核高基重大專項承接企業，也是“兩網一站四庫十二金”國家重點工程的基礎平台提供商，產品廣泛應用於政府、軍工、金融、能源等關鍵行業，累計服務客戶總數超過10萬家。



雲計算國家標準制定企業
金蝶集團旗下基礎軟件企業
資訊技術應用創新核心企業
官網：www.apusic.com

